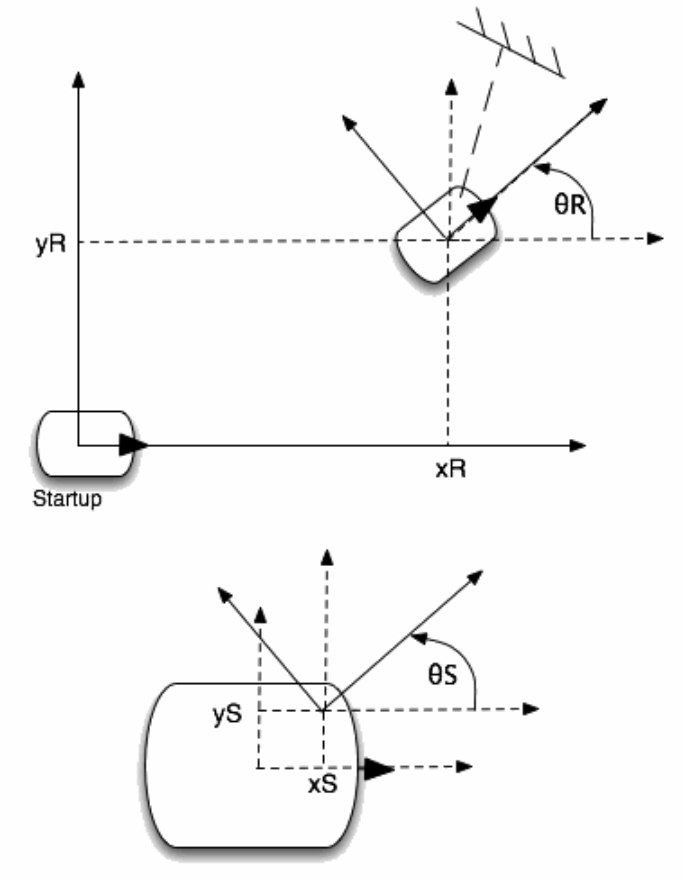
Problem Wk.11.1.6: Sonar hit

One computation that comes up when interpreting the sonar sensors on the robots is the following: Given a distance measurement from one of the sonars, return an instance of `Point` (look at the documentation of the `util` module), in the global odometry frame, the same coordinate frame that the robot's pose is measured, representing where the sonar beam bounced off an object. To do this, we make the assumption that the sonar beam is a line segment emanating from the sonar sensor.

To compute this, we need to know the location and orientation (an instance of `Pose`) of the sonar on the robot and we need to know the Pose of the robot.

- The sonar location on the robot is given by `Pose(xS, yS, thetaS)` where, `xS` and `yS` are the center of the sensor **(relative to the center of the robot)** and `thetaS` is the angle that the beam makes to the robot's heading (the direction the robot's nose points to).
- The robot's pose is `Pose(xR, yR, thetaR)`, as described in the Lab Infrastructure Guide.



It is useful when computing this to think about first finding the location of the hit point relative to the robot and then computing the position of that point relative to the global odometry frame.

Here's a useful bit of math. Imagine you have two coordinate frames, call them A and B. The origin of B is at location (xB, yB), relative to A, and the x-axis of B is rotated by thetaB relative to the x-axis of A. Then, if we we have a point with coordinates bx and

by relative to coordinate frame B, we can find the coordinates ax and ay of that point relative to A as follows:

ax = xB + cos(thetaB)*bx - sin(thetaB)*by
ay = yB + sin(thetaB)*bx + cos(thetaB)*by

Note that when thetaB is zero, this says that ax = xB + bx and ay = yB + by, which is what we would expect. Look at the documentation for `Pose.transformPoint` in module `util`; it transforms the Point by displacing it by pose.x and pose.y and rotating it by pose.theta.

For debugging, you might find it useful to draw a picture of the test cases so as to understand what the answer is supposed to be.

Write the function `sonarHit` that is given a distance measurement from one of the sonars, the sonar's pose on the robot and the robot's pose. It should return an instance of `Point` (in the global odometry frame, the same coordinate frame that the robot's pose is measured) representing where the sonar beam bounced off an object.

You need to specify `util.` to get functions and classes from `util`.

```
def sonarHit(distance, sonarPose, robotPose):
   pass
```